

IDML-APGenerator

IDML-APGenerator

Principes généraux

Fonctionnement divers

Fonctionnement par modification de fichiers modèle

Annexe

`$ID/`

Unités de mesure

Espace dans valeur de propriétés

Référence à un élément imbriqué

Principes généraux

- On ne refait jamais tout. Quand un dossier IDML a déjà été préparé (notamment avec les gros fichiers de ressources), on le laisse intact pour se concentrer uniquement sur ce qui a changé. En général, ce sont les stories, les spreads, les objets.

Fonctionnement divers

Fonctionnement par modification de fichiers modèle

Il y a tellement de propriétés pour que le fichier soit valide qu'on a décidé de fonctionner par fichiers modèles qui seront modifiés. Ils sont comme la « configuration d'usine » du document. La recette se contente donc de modifier au besoin les valeurs différentes.

Pour accélérer la fabrication des fichiers, on prend les fichiers modèle du dossier `lib/assets/models/` et on les transforme sans les parser intégralement, en streaming, en faisant des recherches sur les balises et les attributs.

Une immense table (qui est alimentée au besoin) donne les indications sur le fait qu'une propriété est un `#text` de nœud ou un attribut, et peut préciser les valeurs qu'elle peut recevoir. Elle sert autant à valider les demandes de l'utilisateur qu'à indiquer où et comment modifier la valeur. Voir le fichier `DATA_PROPS.ts` et la table `DATA_PROPS`.

Annexe

`$ID/`

Ce signe, dans les codes IDML/XML, indique que le string qui suit est un *key string*, une clé-string, c'est-à-dire un nom, un texte, qui devra être interprété, à l'importation, suivant la locale définie (la langue).

Pour la production de documents destinés à Affinity Publisher, cette marque est donc inutile, *a priori*.

Unités de mesure

Sauf précision contraire, toutes les valeurs nombres sont exprimées en **points** (le point étant lui-même le 72e d'un pouce — donc `pt * 72 = inch`).

Espace dans valeur de propriétés

Si une espace doit être utilisée dans une valeur de propriété, elle doit être remplacée par `%20` pour que la valeur ne soit pas confondu avec une liste.

```
1 Attr="0 0 0 100" => [0, 0, 0, 100]
2 Attr="0%{20}0%{20}0%{20}100" => "0 0 0 100"
3 # ou
4 Attrs="C'est%20bon"
```

Référence à un élément imbriqué

En règle général, on fait référence à un élément **non imbriqué** par :

```
1 <... Reference="Parent/Self-enfant"...>
```

Par exemple :

```
1 <... AppliedParagraphStyle="ParagraphStyle/MonStyle"...>
```

Mais lorsque la cible est un sous-élément, donc un **élément imbriqué**, on utilise des %30a (« : ») pour marquer le nœud :

```
1 <... ReferenceA="Parent/Sousparent%30aSelf-enfant"...>
```

Par exemple :

```
1 <... AppliedParagraphStyle="ParagraphStyle/List%30aMaList"
```